Improving Factual Inconsistency Detection in LLMs and AlignScore

Alex Chandler
Department of Computer Science, The University of Texas at Austin

1 Abstract

Recent advancements in natural language processing have seen the development of increasingly complex and powerful models, capable of generating text that is both fluent and coherent. However, systems employing large language models are prone to factual inconsistencies when generating text. This research enhances AlignScore, a leading model for evaluating textual consistency. First, we refine AlignScore's methodology by introducing new decomposition techniques for passing in summaries and context into the AlignScore model. We show that breaking down the document context into overlapping intervals for input into AlignScore leads to slight performance gains.

Finally, we assess the capabilities of Large Language Models (LLMs) like GPT to measure factual consistency. Our LLM prompts significantly outperform other language model approaches in evaluating factual consistency on the AggreFact SOTA Test benchmark and the HaluEval Summarization dataset.

2 Introduction

The rise of state-of-the-art Large Language Models (LLMs) such as GPT-3.5, GPT-4, Llama 2, and Claude 2 presents a compelling challenge: while linguistically coherent, their outputs often harbor factual inaccuracies — a phenomenon known as 'hallucination.' While human evaluation remains an essential tool for assessing the quality of these outputs and identifying hallucinations, it is constrained by factors such as financial cost, temporal limitations, and the need for specialized domain knowledge.

Traditional automatic evaluation methodologies like ROUGE (Lin, 2004), METEOR (Banerjee and Lavie, 2005), and BLEU (Papineni et al., 2002) have been instrumental in assessing Natural Language Generation tasks.

However, these metrics display week correlation with human judgments when measuring the factuality of generated text. In particular, these models struggle when there is a significant token-length discrepancy between the reference text and the text they generate for summarization or question-answering tasks.

Consequently, specialized models (Laban et al., 2021; Kryściński et al., 2019; Goyal and Durrett, 2021) have been developed to assess textual factual consistency, verifying the truthfulness of a claim or summary based on a given ground truth text.

However, these factuality assessment models face limitations when tasked with measuring factual consistency across a diverse array of natural language structures including, but not limited to, Natural Language Inference (NLI) and Question Answering (QA) in out-of-domain topics (Tang et al., 2023). The majority of factuality assessment models are trained on relatively homogeneous datasets, where the claims or summaries under evaluation are generated from a singular source or model. This lack of diversity in training datasets limits a trained model's ability to accurately identify the broad spectrum of factuality errors in out-of-domain datasets. (Tang et al., 2023) shows that existing models that measure factual consistency perform significantly worse on generated summaries produced from SOTA models.

In light of these limitations, this study evaluates benchmarks exclusively featuring summaries from state-of-the-art models. This approach more accurately reflects actual usage scenarios of factual consistency, where users more commonly need to validate texts generated by newer LLMs rather than those from older text generation models. We focus on modifying and improving the AlignScore model, as the authors of AlignScore claim it achieves state-of-the-art performance in a variety of natural language tasks, making it a promising candidate for further enhancement in our study.

Existing efforts with using LLMs to identify factual inconsitencies includes (Wang et al., 2023) which prompts the model to return a numerical factuality score from 1-100 and (Luo et al., 2023) to output binary factuality scores of summarization using both Zero-Shot and CoT (Wei et al., 2022) techniques.

AlignScore (Zha et al., 2023) is a state-of-the-art model designed for the automated evaluation of factual consistency in generated text. AlignScore reportedly delivers SOTA performance in assessing factuality on a wide array of natural language tasks, with SOTA performance on common benchmarks including SummaC and TRUE. The model's standout feature is a unified alignment function that enables training on a diverse mix of 15 different datasets, covering 7 major NLP tasks like Natural Language Inference (NLI) and Question Answering (QA). The diverse dataset of 4.7M training examples equips the model with robust generalizability across multiple tasks and domains. Built on a fine-tuned RoBERTa architecture, it incorporates three additional linear layers to perform alignment label prediction through three distinct schemes: binary, ternary classification, and regression.

Formally we define AlignScore's calculation of factual coherence between a claim and a context as:

$$\text{ALIGNSCORE} = \frac{1}{N}\sum_{i=1}^{N}\max_{j=1}^{M}\text{alignment}(s_i, c_j) \tag{1}$$

Where:

- N is the total number of sentences in the generate summary.
- M is the total number of chunks c_j in the context.
- FactualCoherenceScore(s_i, c_j) is the score between claim sentence s_i and context chunk c_j, computed by the fine-tuned RoBERTa model.

AlignScore outputs a numerical score from 0 to 1, where a threshold mapping the numerical score to a boolean factual consistency annotation can learned in the train or validation section of a dataset. As of this study, AlignScore reports the highest Pearson correlation coefficients on human-annotated factual consistency datasets compared to all other previous SOTA models. However, there is room for refinement in its current approaches to claim decomposition and context segmentation techniques.

Specifically, AlignScore performs sentence-level decomposition of claims, breaking them down into "subclaims", where each subclaim corresponds to a single sentence. However, rule-based claim decomposition into sentences potentially prevents the fine-tuned model from focusing computational resources on individually assessing each subclaim that might exist within an individual sentence. (Kamoi et al., 2023) proposes introduce CLAIM-SPLIT, a method of decomposing a claim into subclaims using few-shot prompting with GPT-3.5 Turbo API calls. Building upon CLAIM-SPLIT, we introduce the CLAIM-SPLITS-SELECT method for generating subclaims.

We generate multiple decompositions of a factual sentence using few-shot prompting via GPT-3.5 Turbo API calls, then evaluate each set of subclaims based on factual consistency and completeness. The decomposition with the highest aggregated score is then selected and then fed with each subclaim as its own sentence into the AlignScore model. Regrettably, our methods of claim and context decomposition slightly diminish AlignScore's performance on our benchmarks, likely due to hallucination errors in LLM's claim decomposition attempts.

Additionally, dividing the context into non-overlapping chunks risks severing vital links between adjoining segments, potentially omitting essential contextual cues needed for accurate factual assessment. To mitigate this, our proposal includes the incorporation of overlapping intervals during context segmentation, aimed at preserving critical information and enhancing the model's opportunity to detect factual consistencies. We show that using overlapping intervals during context segmentation as input to AlignScore leads to slight performance gains.

We assess factual consistency using the AggreFact SOTA CNN/DM and AggreFact SOTA XSUM test sets (Tang et al., 2023), benchmarks comprising of SOTA LLM-generated summarizations known for factual inconsistencies that challenge existing models. We additionally test on the Halu-Eval Summarization dataset (Li et al., 2023), a dataset designed by LLMs to be hard to identify factual inconsistencies on.

Finally, we introduce new LLM prompts specifically designed to detect hallucinations and fac-

tual inconsistencies in generated summaries. Our prompts significantly outperform the originally proposed LLM Prompt 10.2in (Li et al., 2023), achieving a significantly increased balanced accuracy on the Halu-Eval Summarization dataset.

3 Datasets

The AggreFact SOTA benchmark (Tang et al., 2023) is specialized for evaluating the factual consistency of generated text summaries. It consists of three main elements: a summary, a context, and a label indicating whether the summary contains any factual inconsistencies that are not supported by the context. The dataset is derived from an aggregation of nine existing annotated factuality datasets, where existing scores for each dataset are initially binary or converted from different annotation schemes to binary.

The AggreFact benchmark is categorized based on the development timeline of the underlying summarization models into FTSOTA, EXFORMER, and OLD categories. The AggreFact dataset, comprising summaries from CNN/DM (Hermann et al., 2015) and XSum (Narayan et al., 2018) articles, is divided into two subsets: AggreFact CNN/DM and AggreFact XSUM. AggreFact XSUM, with its more abstractive summarizations, has a higher error count (Tang et al., 2023), while CNN/DM's more extractive nature results in minor deviations that often lead to factual inconsistency annotations. We use the existing AggreFact SOTA validation and test splits from (Tang et al., 2023) in our work, and use the test subset for all AggreFact benchmarking.

HaluEval (Li et al., 2023) is a comprehensive benchmark tailored for assessing hallucination in LLMs. It utilizes a ChatGPT-based two-step framework, specifically a sampling-then-filtering method, to generate its samples. Our focus is on detecting factual inconsistencies in summarization, so we utilize the summarization subset of Halu-Eval. In the Halu-Eval summarization subset, each of the 10,000 sampled document contexts is paired with two summaries: one with hallucination and one without. (Li et al., 2023) explored the proficiency of Large Language Models (LLMs) in identifying hallucinations via in-context learning, specifically by prompting these models to detect and report hallucinations in summaries. They tested ChatGPT, Claude, Claude2, and Llama, with ChatGPT achieving the hightest accuracy a meager 58.53% in correctly identifying samples with hallucinated content.

4 Benchmarking

We evaluate AlignScore and our LLM prompts on both the AggreFact benchmarks on XSUM and CNN/DM test datasets as well as on the HaluEval-Summarization dataset. To cut computing costs, we limit our sample size for LLM prompt testing, as denoted by *, and reserve our AlignScore methodology modifications to the AggreFact-SOTA XSUM and CNN/DM test benchmark. We use balanced accuracy to evaluate the performance of factuality metrics due to the imbalance of factually consistent and inconsistent summaries in the AggreFact benchmark. Additionally, we measure and report sensitivity and specificity.

4.1 AggreFact Benchmarking

As a reminder, AlignScore outputs a numerical score from 0 to 1, where 0 represents a claim or summary that is not supported by the input context. To apply AlignScore to AggreFact, we learn a threshold value of AggreFact SOTA XSUM and AggreFact SOTA CNN/DM initially on the validation subsets of both datasets respectively, optimizing the threshold for the highest balanced accuracy threshold. The learned threshold value for optimizing balanced accuracy on AggreFact SOTA CNN/DM Validation sets are 0.9149 and 0.7818 on AggreFact SOTA CNN/DM XSUM. The results for AlignScore on AggreFact SOTA Test when learning the linear thresholds on AggreFact Validation can be seen in Table 1.

Dataset	Bal. Acc.	Sens.	Spec.
AggreFact SOTA XSUM	58.16	20.35	95.97
AggreFact SOTA CNN/DM	61.63	39.04	84.21

Table 1: Performance Metrics for AggreFact SOTA XSUM and CNN/DM Test sets with linear threshold learned on Validation set

Note, how the balanced accuracy and sensitivity values are significantly lower than when we set the optimal binary threshold based on the same dataset we are testing. If we optimize our learned binary thresholds off of AggreFact SOTA Test CNN/DM and AggreFact SOTA Test XSUM, our learned threshold of AggreFact SOTA XSUM are 0.7467

and 0.2843 respectively, leading to the much improved results in Table 2. This observation suggests that AlignScore's performance is highly dependent on the details and error characteristics of the summaries it is assesses factuality on, emphasizing the need for dataset-specific threshold optimization to accurately assess factual consistency in different summarization contexts.

Dataset	Bal. Acc.	Sens.	Spec.
AggreFact CNN/DM	66.70	75.45	57.80
AggreFact XSUM	71.85	72.63	71.06

Table 2: Performance Metrics for AggreFact SOTA CNN/DM and XSUM Test sets with linear threshold learned on Test set

Model	CNN/DM	XSUM
AggreFact	66.70	**71.85**
DAE	59.4^{T}	73.1^{T}
QuestEval	63.7^{T}	61.6^{T}
SummaC-ZS	63.3^{T}	56.1^{T}
SummaC-Cv	**70.3**^{T}	67.0^{T}
QAFactEval	61.6^{T}	65.9^{T}
Luo 2023 ChatGPT-ZS	66.2^{T}	62.6^{T}
Luo 2023 ChatGPT-CoT	49.7^{T}	56.0^{T}

Table 3: Balanced Accuracy Scores for AggreFact SOTA CNN/DM and XSUM Across Encoder Models. Boldened numbers indicate the highest scoring model for AggreFact SOTA CNN/DM and XSUM respectively. Superscript ^{T} denotes reported benchmarking from (Tang et al., 2023)

Since Halu-Eval did not release a train-test-validation split of the Halu-Eval summarization dataset, we split the whole dataset into our own train-test-val subset. Our test subset consists of 500 summaries with factual hallucinations, and 500 summaries without factual hallucinations.

For AlignScore benchmarking on Halu-Eval summarization, we randomly extract a test set of 1000 rows, where 500 are summaries without hallucinations and 500 are summarizations with hallucinations. We set AlignScore's threshold on the remaining 19,000 context-summary combinations in the Halu-Eval summarization dataset, and learn the best threshold for balanced accuracy is 66.6% Our benchmarking for AlignScore on Halu-Eval can be seen in table 4. SOTA 10.2 is the LLM prompt proposed in (Li et al., 2023) to identify hallucinations in summarization and can be seen in 10.2.

Model	Bal. Acc.	Sens.	Spec.
AlignScore	67.6	78.6	57.4
Prompt 10.2	57.4	38.4	76.4
Prompt 10.3	52.5 *	45.0*	60.0*

Table 4: Performance Metrics for Halu-Eval Summarization Test Subset comparing AlignScore to existing LLM Prompt Based Solutions proprosed in (Li et al., 2023) and (Luo et al., 2023). *Values based on a reduced sample size of 200 to minimize computation costs.

4.2 LLM Prompt Benchmarking

Table 5 shows benchmarking for our own prompts, (Luo et al., 2023) CoT prompt, and (Li et al., 2023) hallucination detection prompt across all of our benchmark datasets. We observe that our prompts outperform existing published LLM prompts for detecting factual inconsistencies. However, further analysis would need to be completed on the full AggreFact benchmark and Halu-Eval dataset without sampling to obtain a more precise difference in their performance.

5 Understanding AlignScore Scoring Distribution

When applied to the AggreFact dataset, where claims consist of a one to multiple sentences, AlignScore calculates scores by mapping each claim to individual context chunks. It defines the factuality score of each claim as the maximum factuality score of any subcontext pairing to that score. Then, it takes the average of all maxed factaulity scores for each claim, and outputs 'SUPPORTED' if this averaged maxed factuality scores for all claims exceeds a learned linear threshold. For ease of visualization, we look particularly at a subset of factuality scores for summarizations with only one claim.

Figure 2 shows AlignScore's performance on AggreFact-XSUM-SOTA subset of the test set. Surprisingly, the majority of factual inconsistencies are not detected on AggreFact XSUM SOTA with AlignScore. The learned threshold of .283, while optimizing for balanced accuracy, results in a model that fails to identify most factual inconsistencies. If the end goal of model is to identify

Dataset	*Prompt*	Bal. Acc.	Sens.	Spec.
X_SUM	1	67.2	40.4	93.8
	2	70.2	43.6	96.8
	3	68.4	40.2	96.4
	10.2	57.0*	33.5*	80.5*
	10.3	60.0	35.0	75.0
CNN	1	72.6	58.2	87.4
	2	66.0	34.6	97.2
	3	70.4	59.4	81.4
	10.2	48.0*	44.0*	52.0*
	10.3	57.0*	31.0*	83.0*
Halu-Eval	1	68.2	47.0	89.4
	2	66.8	44.0	89.6
	3	65.8	34.6	97.0
	10.2	57.4	38.4	76.4
	10.3	52.5*	45.0*	60.0*

Table 5: Sensitivity, Specificity, and Balanced Accuracy across AggreFact XSUM SOTA Test, AggreFact CNN/DM SOTA Test, and Halu-Eval Summarization Test for our three proposed Prompts 10, the CoT prompt from (Luo et al., 2023), and the hallucination detection prompt proprosed in the prompt proposed in (Li et al., 2023). *Values based on a reduced sample size of 200 to minimize computation costs.

all factual inconsistencies, then balanced accuracy may not be the best metric to evalaute factual consistency models with an unbalanced dataset. Rather, True Positive to False Positive and True Negative to False Negative ratios may be more meaningful to understand how much you can trust each output of a factual consistency model. Similarly, in 3 optimizing our threshold for balanced accuracy leads to the majority of factual consistencies not being detected.

6 Methodology

6.1 LLM Prompt Design

The design of our LLM prompts was conducted using GPT-4 in the OpenAI playground for creation and GPT-3.5 Turbo for testing. The prompts were crafted to embody a Chain of Thought (CoT) approach, each guiding models through a structured thought process for evaluating factual consistency. Key elements of our approach included explicit evaluation criteria and a focus on inference verification, where models determined if each claim's information could be inferred from the article. Recognizing that existing LLM prompts often missed factual inconsistencies and hallucinations, we introduced stricter criteria, directing the model to label a summary as 'SUPPORTED' only if every detail aligns with the article's content. Any detail not explicitly supported or falling under an error category leads to a 'NOT SUPPORTED' classification. We share our prompts in the appendix.

6.2 CLAIM-SPLITS-SELECT

Building upon CLAIM-SPLIT (Kamoi et al., 2023), which decomposes factual sentences into subclaims using few-shot prompting with the GPT-3 API, we introduce an enhanced methodology called CLAIM-SPLITS-SELECT. This approach consists of two pivotal steps. First, we generate multiple sets of subclaims using few-shot prompting via the GPT-3 API. Second, we select the best set of subclaims by exploring various objective functions, each employing a distinct method or model for determining the most complete and factually accurate decomposed claim. The set of subclaims that attains the highest aggregate score is selected as the set of claims to verify by the AlignScore model. The overall factuality score of the original claim is then computed as the mean of these individual verification scores.

6.3 Step 1: Generating Sets of Sub-claims

The initial phase in our methodology, as well as in the CLAIM-SPLIT method (Kamoi et al., 2023), leverages a GPT-based language model to generate multiple sets of subclaims that decompose an original factual claim. The CLAIM-SPLIT approach utilizes an instruct-tuned GPT-3 model (text-davinci-002) and guides the model using a prompt, "Segment the following sentence into individual facts," supplemented with six example claim splits. The aim is to provide sub-claims that cover the entire information content of the original claim.

Our approach, CLAIM-SPLITS-SELECT, builds upon this by introducing several enhancements:

- **Dynamic Examples:** We augment the fixed examples used in CLAIM-SPLIT with dynamically chosen examples for the prompt. To achieve this, all examples from CLAIM-SPLIT, as well as additional ones where a claim is a single sub-claim, are vectorized. The purpose of including single sub-claim examples is to minimize the risk of inappropriately dividing such claims into multiple sub-claims. We employ a 'SemanticSimilar-

ityExampleSelector‘ with HuggingFaceEmbeddings and Chroma for vector storage to measure semantic similarity among these vectorized examples. The selector identifies the k most semantically similar examples to the claim in question, where $k = 6$, identical to CLAIM-SPLIT. These selected examples are then used in the prompt for sub-claim generation.

- **Diversity in Generations:** To enhance the diversity of generated subclaims, we progressively increase the temperature parameter during successive GPT-3 API calls for each new decomposition attempt of the same claim. This adjustment allows for a broader range of potential subclaims in the generated sets.

- **Multiple Sets:** Unlike CLAIM-SPLIT which produces one set of sub-claims, our approach generates three sets of subclaims for each original claim. This enriched generation step results in each original claim c being decomposed into multiple sets of subclaims $\{c1, c2, \ldots, cm\}$, where each set is a candidate for subsequent evaluation and selection steps.

6.4 Step 2: Selecting Best set of Sub-claims

We describe two main methods for selecting the most suitable set of subclaims among the generated candidates.

Our Select Phase aims to score and select the best set of subclaim based on the following criteria:

- **Completeness**: This metric gauges the extent to which the set of subclaims spans the informational content of the original claim. It evaluates how well the subclaims collectively represent all facets of the original statement.

- **Accuracy**: This metric evaluates how closely each subclaim adheres to the factual and contextual elements of the original claim. It measures the fidelity of each subclaim to the actual information presented.

Each of the following two methods for aim to select the most accurate and complete set of subclaims.

1. **GPT-Select**

   - Utilizing GPT-3.5, we prompt the model to internally evaluate each set of subclaims for both completeness and accuracy, and then have it return the set with the highest aggregate score, effectively bypassing the need for an external objective function.

2. **Hybrid-Select**

   - CompletenessScore: We prime GPT-3.5 to score the completeness of each set of subclaims.
   - Accuracy Score: We employ AlignScore to score the factual consistency of each sub-claim as it relates to if it can be supported by the entire claim, and output an overall accuracy score that is the mean of the factual consistency scores for each sub-claim.
   - The aggregate score employed to assess each set of subclaims is formulated as a weighted sum, comprising the GPT-generated completeness score (C_{GPT}) and the AlignScore accuracy measure ($A_{\text{AlignScore}}$), with learned weights α and β. These weights are optimized to achieve the highest balanced accuracy on a validation set. **Objective Function for HybridAggregation-LLM-Align**: $\text{Objective}_{\text{HybridAggregation-LLM-Align}} = \alpha \times C_{\text{GPT}} + \beta \times A_{\text{AlignScore}}$
   - **Completeness Scoring Variants**:
     - **BatchScoring**: GPT evaluates the completeness of each batch of generated subclaims collectively, returning a list of completeness scores by querying the model for all sub-claim decomposition attempts only once for each claim to return a set of completeness cores.
     - **IndividualScoring**: GPT evaluates the completeness of each individual subclaim within a set, where we must query GPT for each set of sub-claims individually.

6.5 Overlapping Context Segmentation

In this work, we introduce an algorithm named *ContextSegmenter* that is tailored for dividing a textual context into coherent sub-contexts. The algorithm aims to achieve two primary objectives:

1. To ensure that each sentence is represented multiple times across different sub-contexts, with the frequency of inclusion effectively controllable based on the algorithm's parameters.

2. To maintain contextual cohesion between adjacent sub-contexts by enforcing a minimum token overlap.

The algorithm is controlled by two hyperparameters, which are determined empirically:

- α : The maximum allowable token count for any individual sub-context.
- ω : The minimum number of tokens that must overlap between adjacent sub-contexts.

Operational Steps

1. The entire context is partitioned into its constituent sentences.

2. Sentences are cumulatively aggregated until the token count approximates, but does not surpass α. This forms the initial sub-context. Any sentence that would make the token count exceed α is reserved for the succeeding sub-context.

3. For each subsequent sub-context, sentences are incorporated until nearly α tokens are reached, and ensuring a minimum of ω tokens overlap with the preceding sub-context. The start point for each new sub-context is adapted to the closest sentence boundary, effectively rounding by sentence to meet the ω-token overlap requirement.

By this approach, *ContextSegmenter* generates sub-contexts that are not only self-contained but also have a meaningful overlap with adjacent sub-contexts. This mechanism aids in preserving the contextual continuity and richness of the original textual input. Additionally, it gives the model multiple chances at identifying any factual inconsistency between any claim and part of the context.

6.6 AlignScore Modification Results

In Table 6, we observe that only the Overlapping Context methodology results in improved performance on AggreFact XSUM SOTA and AggreFact CNN/DM SOTA benchmarks. We see a 1.3 % and 2.6 % increase in balanced accuracy when using our overlapping context methodology across XSUM and CNN/DM respectively. Surprisingly, all claim decomposition approaches failed to enhance accuracy, and some techniques even led to comparable or reduced performance. We hypothesize that utilizing GPT 3.5 for claim decomposition introduces errors that do not justify the theoretical performance gains achieved by allowing each sub-call to AlignScore to focus on evaluating the factual consistency of individual claims.

7 Limitations

In the Limitations section of our paper, we recognize several constraints in our methodology and results. We did not evaluate AlignScore on the entire Halu-Eval dataset, which may limit the generalizability of our findings. Additionally, our experiments maintained a constant temperature setting of 0.3, not exploring the impact of varying this parameter. While we observed a 2-4% increase in balanced accuracy when using GPT-4 on a small sample size on the AggreFact SOTA benchmarks, we did not comprehensively test more powerful models like GPT-4 across a large enough sample size to include it in this study. We recognize that despite our claim-decomposition techniques not enhancing performance, a broad spectrum of decomposition techniques remains unexplored, which could potentially yield significant insights into performance enhancement.

8 Analysis and Future Work

Our research into factual consistency assessment opens several avenues for further study. Developing a wider range of LLM prompts could provide a more comprehensive evaluation of factual accuracy. Investigating weak supervised learning methods for prompt ensembling may lead to stronger, more reliable results. A thorough error analysis of claim decomposition is necessary to understand why current methods are not yielding improvements. Exploring different score ensembling techniques could also enhance model performance. Further, a deeper error analysis of AlignScore is crucial, as our results suggest it may not be as effective in generalizing and detecting factual inconsistencies in state-of-the-art LLM-generated text as previously thought. Expanding tests of LLM prompts to larger dataset subsets would provide a more robust evaluation.

Methodology	XSUM Balanced. Acc	CCC/DM Balanced. Acc
Default	71.8	66.6
Overlapping Context	73.1	69.2
Decomposed Claims with CLAIM-SPLIT	70.3	65.6
Decomposed Claims with CLAIM-SPLITS-SELECT	71.5	.68.0

Table 6: Performance metrics for AlignScore on AggreFact SOTA XSUM and CNN/DM benchmarks when modifying model methodology.

9 Conclusion

In this study, we introduced new state-of-the-art prompts designed for more accurately evaluating factual consistency, demonstrating their effectiveness on the challenging AggreFact SOTA benchmarks and Halu-Eval Summarization dataset. In this work, we present a method to improve the detection of factual consistencies with AlignScore by segmenting text into overlapping subcontexts, ensuring complete coverage and contextual cohesion between adjacent segments. However, our overall findings highlighted limitations in the AlignScore model, particularly in its performance on the AggreFact SOTA test benchmarks. When setting the score threshold on the validation test set, AlignScore struggled to consistently detect factual inconsistencies in text generated by recent text generation models.

10 Appendix

10.1 Our Prompts

Prompt 1

Assess the factual consistency of the following claim based solely on the information provided in the summary. Do not make inferences or assumptions beyond the provided information.
Summary: {context} Claim: {summary}
Assessment: [Think step by step. Use the following chain of thought. Step 1: Remember the claim. Step 2: Remember the summary. Step 3: Break down all the lines in the claim and remember them. Pay close attention to numbers associated with the events and store them separately. Step 4: Follow the given chain of thought. After following these steps, decide if the claim is SUPPORTED or NOT SUPPORTED. Be sure that you output your answer as SUPPORTED or NOT SUPPORTED

Prompt 2

Evaluate the claim below based on the context provided in the article. Your evaluation should consider the following:
1. Does the article explicitly state the information contained in the claim? 2. Can the information in the claim be inferred from the article even if not explicitly stated? 3. Is there any information in the article that contradicts the claim? 4. Does every detail in the claim find support in the article's context?
In light of the above points, determine if the article supports the claim.
Article: {context} Claim:{summary}
Answer (SUPPORTED or NOT SUPPORTED)

Prompt 3

In this task, you are required to meticulously analyze the factual consistency of a summary against the original article. Your focus should be on validating every single detail presented in the summary, regardless of its perceived significance.
Procedure:
1. Break Down the Summary: - Identify and list each specific detail and claim made in the summary.
2. Rigorous Verification: - For each detail listed, cross-reference it with the information in the article. - Pay close attention to minor details and subtle nuances.
3. No Room for Inference: - Avoid making inferences or assumptions if the information is not explicitly stated in the article. - The summary must be factually consistent with the article in every aspect.
4. Specific Error Type Analysis: - For each category of error (Negation, Adjective, Coreference, etc.), specifically ask if

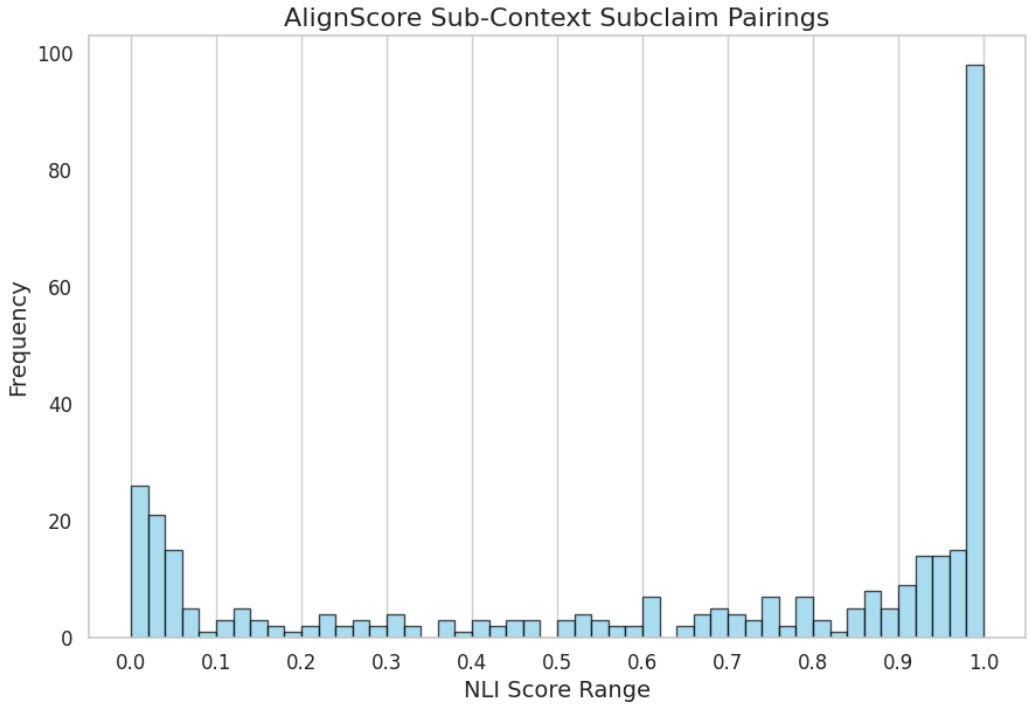

Figure 1: This histogram visualizes the AlignScore output for sub-context and sub-claim pairings on a combination of AggreFact CNN/DM and AggreFact XSUM SOTA. The distribution is notably bimodal, with scores primarily clustered close to 0 or 1, indicating a polarization in the model's confidence. This pattern implies that AlignScore is decisive in either strongly supporting or refuting sub-claims based on the given sub-contexts.

such errors are present in the summary. - Analyze each detail for potential Negation, Adjective, Coreference, Number, Entity, Attribute, Pronoun, Commonsense, Temporal, Predicate, Discourse Link, Relation, Quantity, Event, Noun Phrase, and Circumstance Errors.
5. Strict Criteria for Support: - Only classify the summary as 'SUPPORTED' if every single detail aligns with the article's content, considering the specific error types. - If even one detail is not explicitly supported by the article or falls under any error category, classify the summary as 'NOT SUPPORTED'.
Article: {context} Summary: {summary}
Answer (SUPPORTED or NOT SUPPORTED):

10.2 Existing Halu-Eval SOTA Prompt

Halu-Eval Proposed Prompt For Identifying Factual Consistencies

Follow the instruction. Instruction: Determine if the text is consistent or inconsistent with the provided knowledge and dialogue history. If there is a logical conflict, respond with 'inconsistent'. If there is no conflict, respond with 'consistent'. Input: Text: {context}. Dialogue History: {summary}: Please summarize the given knowledge. Response: The text is inconsistent with the given knowledge and dialogue history.

10.3 Luo 2023 Prompts

Luo 2023 Zero-Shot Prompt

Decide if the following summary is consistent with the corresponding article. Note that consistency means all information in the summary is supported by the article. Ar-

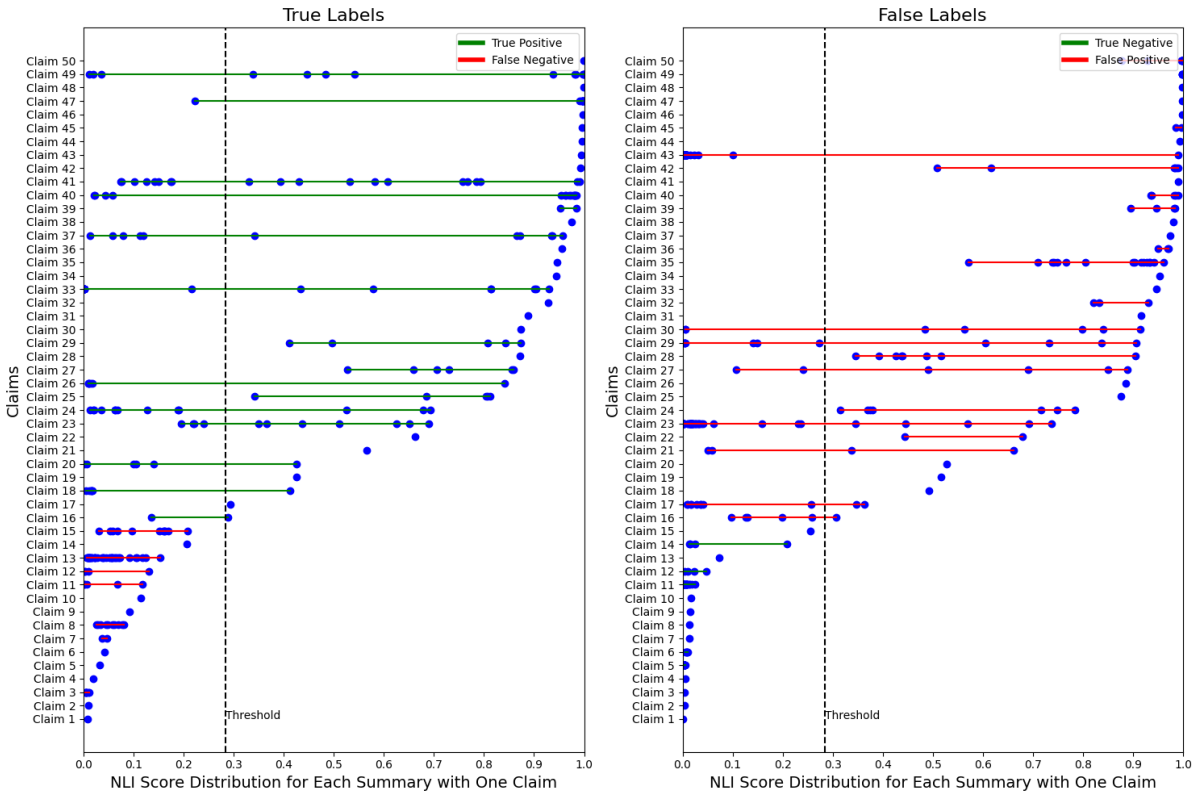

Figure 2: This plot visualizes the sampled distribution of natural language inference (NLI) scores generated by AlignScore on AggreFact XSUM-SOTA, categorized by True and False labels. Each lines represents the range of AlignScores model output from all specific claim sub-context pairings from summaries with only one claim. In other words, each line represents the factuality scores from a different summary of the dataset. Individual circles mark the exact scores produced by a specific claim - sub-context pairing. Green lines signify instances where AlignScore accurately assessed the overall factuality of the claim, while red lines highlight cases where the claim was incorrectly assessed factuality.

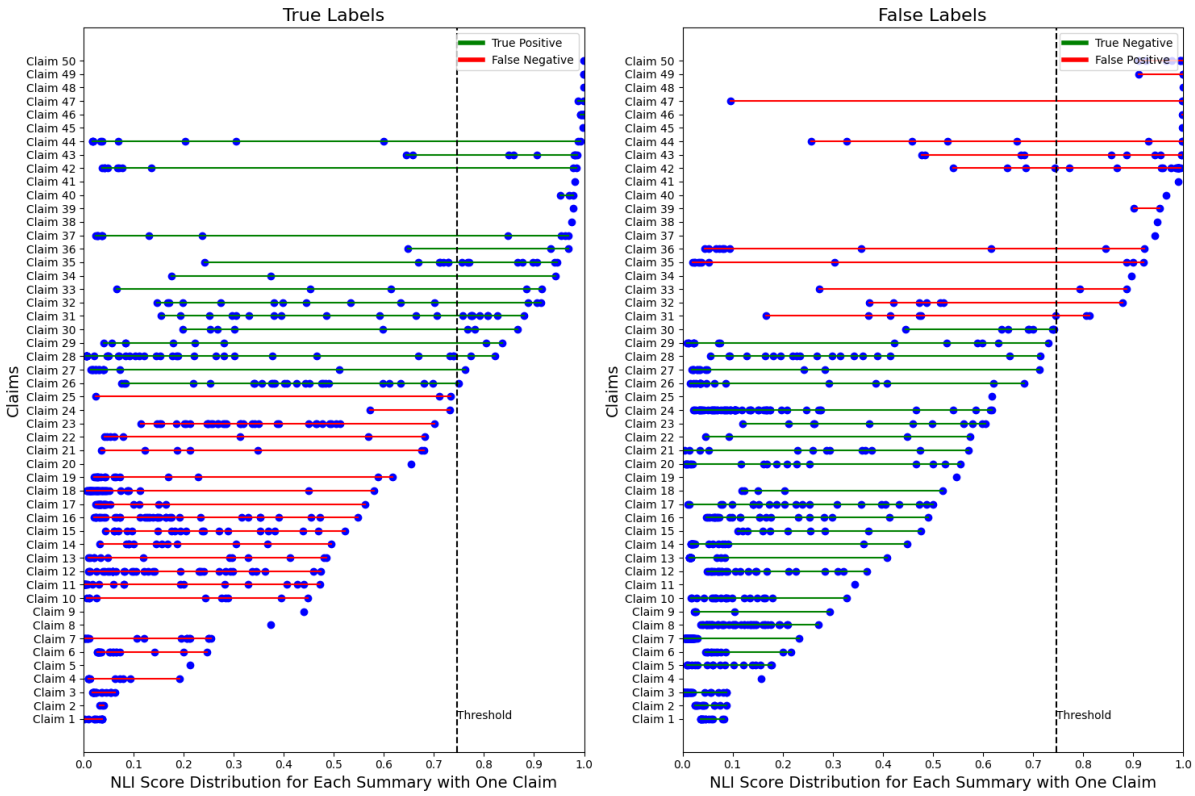

Figure 3: This plot visualizes the sampled distribution of natural language inference (NLI) scores generated by AlignScore on AggreFact CNN/DM-SOTA, categorized by True and False labels. Each lines represents the range of AlignScores model output from all specific claim sub-context pairings from summaries with only one claim. In other words, each line represents the factuality scores from a different summary of the dataset. Individual circles mark the exact scores produced by a specific claim - sub-context pairing. Green lines signify instances where AlignScore accurately assessed the overall factuality of the claim, while red lines highlight cases where the claim was incorrectly assessed factuality.

ticle: [Article] Summary: [Summary] Answer (yes or no):

Luo 2023 Chain-of-Thought Prompt breakable

Decide if the following summary is consistent with the corresponding article. Note that consistency means all information in the summary is supported by the article. Article: [Article] Summary: [Summary] Explain your reasoning step by step then answer (yes or no) the question:

References

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Tanya Goyal and Greg Durrett. 2021. Annotating and modeling fine-grained factuality in summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend.

Ryo Kamoi, Tanya Goyal, Juan Diego Rodriguez, and Greg Durrett. 2023. Wice: Real-world entailment for claims in wikipedia.

Wojciech Kryściński, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Evaluating the factual consistency of abstractive text summarization.

Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. 2021. Summac: Re-visiting nli-based models for inconsistency detection in summarization.

Junyi Li, Xiaoxue Cheng, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023. Halueval: A large-scale hallucination evaluation benchmark for large language models.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Zheheng Luo, Qianqian Xie, and Sophia Ananiadou. 2023. Chatgpt as a factual inconsistency evaluator for text summarization.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation.

Liyan Tang, Tanya Goyal, Alexander R. Fabbri, Philippe Laban, Jiacheng Xu, Semih Yavuz, Wojciech Kryściński, Justin F. Rousseau, and Greg Durrett. 2023. Understanding factual errors in summarization: Errors, summarizers, datasets, error detectors.

Jiaan Wang, Yunlong Liang, Fandong Meng, Zengkui Sun, Haoxiang Shi, Zhixu Li, Jinan Xu, Jianfeng Qu, and Jie Zhou. 2023. Is chatgpt a good nlg evaluator? a preliminary study.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Yuheng Zha, Yichi Yang, Ruichen Li, and Zhiting Hu. 2023. Alignscore: Evaluating factual consistency with a unified alignment function.